Spezielle Algorithmen

P, NP und NP-Vollständigkeit

Komplexitätsklassen zur Einteilung von Problemen in Schwierigkeitsgraden bzw. Lösungswegen.

P Probleme

Umfasst alle praktisch lösbaren Probleme, die in höchstens n^k Zeiteinheiten gelöst werden können. In polynomieller Zeit *deterministisch* lösbare Probleme.

NP Probleme

In polynomieller Zeit *nicht deterministisch* lösbare Probleme, gelten als praktisch unlösbare Probleme.

Probleme können durch *Guess-and-Check*, also Raten einer Lösung und Verifizierung der Lösung, gelöst werden.

Reduktion von NP Problemen

2 Entscheidungsprobleme *A* und *B* gegeben. Polynomieller Algorithmus für *B* existiert. Eingabe von *A* wird in Eingabe von *B* umgebaut ⟶ Ausführung von *B* mit umgebauter Eingabe führt zu Ergebnis von *A*.

$A \leq_p B$

> **Satz von Cook-Levin**: Eine SPrache *L* heißt NP-schwer, wenn sich jede Sprache *L*' aus NP in deterministisch polynomieller Zeit auf *L* reduzieren lässt.

⇒ Jedes Problem in NP lässt sich in Polynomialzeil auf SAT-Problem reduzieren, SAT ist NP-vollständig

Sei *L* **NP-vollständig**, dann gelten:

- $L \in P \Leftrightarrow NP \subseteq P$
- $L_2 \in NP$ und $L \leq_p L_2 \Rightarrow L_2$ ist NP-vollständig
- L_2 ist NP-vollständig $\Rightarrow L \equiv L_2$

3-COLOR

3-COLOR kann zur Lösungsfindung eines 3-SAT-Problems genutzt werden. Beim 3-COLOR Problem dürfen 2 benachbarte (also durch eine Kante verbundene) Knoten nicht in der selben Farbe eingefärbt sein.

3-COLOR zur Lösung von 3-SAT

1. Aus jedem Literal (jeder Variable x_n und $\bar{x}_n$) wird ein Knoten

2. Knoten T(rue), F(alse) und B(lue) werden mit allen Literalen verbunden
3. Jedes Literal wird mit seiner Negation verbunden
4. Für jede Klausel (Verknüpfung von 3 Literalen) des 3-SAT Problems wird ein Hilfsgraph mit 6 Knoten und 13 Kanten eingezeichnet
5. Literale werden entweder True oder False, also Grün oder Rot gefärbt
6. Befülle Knoten im Hilfsgraph von oben nach unten mit Farbbelegung zur Prüfung, ob Klausel erfüllbar

Approximationsalgorithmen

Approximationsalgorithmen nähern Wert einer optimalen Lösung in polynomieller Laufzeit an.

Sei *A* ein Approximationsalgurithmus für Optimierungsproblem mit Eingabe *E* und Ausgabe *A(E)*:

Relative Güte: $R_A(E) = \max\{\frac{A(E)}{OPT(E)}, \frac{OPT(E)}{A(E)}\}$

> **Satz**: Falls $NP = P$ ist, gibt es keinen polynomiellen Approximationsalgorithmus für TSP mi konstanter relativer Güte.

TSP - Problem des Handlungsreisenden

Metrisches Problem des Handlungsreisenden ΔTSP beschreibt TSP, bei dem die Gewichtung der Kanten die Dreiecksungleichung ($a + b > c$) erfüllt.

Algorithmus von Prim - Minimaler Spannbaum

Minimaler Spannbaum nach Prim

1. Baum wird an einem beliebigen Knoten gestartet
2. jeweils leichteste Kante zu Knoten, welcher noch nicht im Baum enthalten ist, wird gewählt

Algorithmus von Christofides - Approximation ΔTSP

1. Berechnung minimaler Spannbaum *T*
2. Bestimme alle Knoten *S* mit ungeradem Grad
3. Berechne leichtestes Matching *M* auf *S*
4. Füge Kanten von *M* zu *T* hinzu
5. Berechne Eulertour in $M \cup T$, entferne doppelte Kanten
6. Gib Knotenreihenfolge aus